

Cont
A4

creating a package file which contains information indicating changes in a computer system from an initial state to a modified state;
creating a user profile associated with a user;
storing the package and user profile on a first computer of the distributed data processing system such that the package and user profile can be sent to individual computers of the distributed data processing system;
upon startup of a second computer of the distributed data processing system, allowing access to the package and user profile by the second computer of the distributed data processing system.

29. (NEW) A method of managing a distributed data processing system, comprising the computer implemented steps of:

capturing changes in a computer system between an initial state and a modified state;
separating the changes into user-specific changes and system-specific changes;
wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis.

REMARKS

Claims 1-29 are now pending in the present application. Claims 1, 14, and 27 were amended; and claims 28 and 29 were added. These changes are believed not to add new matter. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 103, Obviousness

The examiner has rejected claims 1, 3-5, 9-11, 13, 14, 16-18, 22-24 and 26-27 under 35 U.S.C. § 103 as being unpatentable over U.S. Patent No. 5,860,012 issued to Luu in view of U.S. Patent No. 6,101,601 issued to Matthews et al. This rejection is respectfully traversed.

As to claims 1, 3-5, 9-11, 13, 14, 16-18, 22-24 and 26-27, the Office Action states:

As to claim 1, the reference of Luu teaches of: Capturing an initial state and a modified state (capturing the state of the LAN Administrator's system before and after the installation of the application, col. 1, lines 66-67, col. 2, lines 1-2); Modifying (modifies system configuration files, col. 8, lines 5-9, instruction for modification of the pre-installation snapshot and post installation files, col. 6, lines 18-31); Storing differences between initial state and the modified state (stored the differences of the pre-installation snapshot and the post-installation snapshot in the installation package, col. 6, lines 21-33); Manage configurability (using the differences, IpackGen creates the Ipack format file, appendix A, col. 11 and col. 12, Ipack format file contains instructions to enable modification, col. 6, lines 26-33, replaceable parameters, appendix A, col. 29).

While the reference of Luu teaches of capturing the state of the system before and after the installation (col. 1, lines 66-67, col. 2, lines 1-2), it fails to explicitly teach of the initializing step. The reference of Matthew et al teaches of initializing a data processing system when the system is booted up (Fig. 6, 602, initializing, col. 2, lines 52-61, col. 6, lines 35-41) for assigning a beginning value to a variable before start up. It is obvious that the system has to be initialized before it can run. Therefore, it would have been obvious for one ordinary skill in the art at the time the invention was made to include the initializing step to Luu's invention for the purpose of starting up the computer and be able to run the programs.

Analysis

Claim 1, amended in this reply, is reproduced for purposes of discussion:

1. (ONCE AMENDED) A method for identifying and storing changes to a data processing system within a distributed data processing system, the method comprising the computer-implemented steps of:

- initializing the data processing system for a capture of an initial state of the data processing system;
- modifying the data processing system;
- capturing a modified state of the data processing system; and
- storing differences between the initial state and the modified state as a set of configuration parameters in a depository, wherein the differences are separated into system-specific changes and user-specific changes; and

wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis.

It is noted that the phrase, "wherein the differences are separated into system-specific changes and user-specific changes; and wherein the system specific-changes are applied

on a per-system basis and the user-specific changes are applied on a per-user basis," has been added to Claim 1 (as well as Claims 14 and 27). It is believed that these limitations of the claims are not shown in any of the references cited against the present application.

These limitations are supported in the present specification at least on page 15 of the specification, which states,

The present invention also addresses the problem of user-specific application files for individual users. By separating the system files from the user files, the user may roam between a series of different workstations and maintain a consistent application environment. The present invention also captures and splits the changes into between user-specific and system-specific changes so that the system changes may be applied per machine and the user changes may be applied per user.

There is no apparent mentioning in any cited reference of separating the changes into system-specific changes and user-specific changes, and applying them on system or user bases. This feature of the present innovations provides the advantage of selective modification of the client machines in the distributed data processing system, and also allows seamless movement between systems while maintaining a consistent user environment both with respect to applications available and individual user settings for the computer work environment.

Since Claims 3-5, 9-11 and 13 depend ultimately from Claim 1, these dependent claims are believed distinguished from the cited references.

Further, the independent Claims 14 and 27 also include the above referenced feature (added by amendment), and are also believed allowable for the same reasons as Claim 1. Therefore, Claims 14, 16-18, 22-24 and 26-27 are believed allowable.

Therefore, the rejection of claims 1, 3-5, 9-11, 13, 14, 16-18, 22-24 and 26-27 under 35 U.S.C. § 103 has been overcome.

Further, dependent Claims 2, 6-8, 12, 15, 19-21, and 25 also depend from the above-mentioned independent claims. Therefore, it is believed that all claims of the present application are allowable over the cited references.

Certain dependent claims are also believed allowable for separate reasons, addressed below.

Regarding Claim 5, which is reproduced here:

5. The method of 4 wherein the snapshot may be configured to include or to exclude portions of data within the data processing system.

In rejecting this claim, the Examiner cites Luu at col. 5, lines 53-59 as teaching, "the snapshot is configured to include portions of data (snapshots contain all the information such as the complete disk directory structure contents, copies of various systems and execution files...)". Col. 5, lines 53-59 are reproduced:

This preinstallation system snapshot will contain all the information that may be changed as a result of the installation of the application software, such as the complete disk directory structure contents, copies of various system and execution files (e.g. the CONFIG.SYS file and all *.INI and *.BAT files) and other operating system related information.

However, this passage states that, "The preinstallation system snapshot will contain all the information that may be changed as a result of the installation of the application software...." This passage does not teach the limitation of Claim 5, i.e., that "the snapshot may be configured to include or to exclude portions of data within the data processing system." [Emphasis added.] Excluding certain data in the context of the present invention is not taught or mentioned in Luu or any other cited reference. Teaching that "all the information" will be included in the snapshot does not encompass or teach exclusion of some of the information.

Hence, Claim 5 is believed distinguished over the cited references and is believed allowable.

The Examiner also rejects Claim 10, saying that, "the reference Luu inherently teaches of capturing line-by-line. It is "well-known" in the art that the capturing step has to be captured line-by-line in order to compare the files for the differences. Therefore the step of capturing line-by-line is inherent in order to compare the differences between the files."

Applicant respectfully requests that Examiner comply with the MPEP, sec. 2112 regarding Burden of Proof for rejections based on inherency. This MPEP section states,

"The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the **inherency** of that result or characteristic." Later, the MPEP quotes caselaw to illuminate the burden Examiner must fulfill:

"In relying upon the theory of **inherency**, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art." *Ex Parte Levy*, 17 USPQ 1461, 1464 (Bd. Pat. App. & Inter. 1990) (emphasis in original)

It is respectfully believed that Examiner has presented no facts or reasoning to support the assertion that there is no other way to compare .INI files other than a line-by-line capture. Without such showing, it is believed that Examiner has failed to make out a prima facie case of obviousness under 35 USC 103.

Claim 10 is therefore believed allowable over the cited references.

The examiner has rejected claims 6-8, 12, 19-21 and 25 under 35 U.S.C. § 103 as being unpatentable over U.S. Patent No. 5,860,012 issued to Luu in view of U.S. Patent No. 6,101,601 issued to Matthews et al. as applied to claims 1 and 14 above, and further in view of U.S. Patent No. 6,029,246 issued to Bahr. This rejection is respectfully traversed.

As to claim 7, the Office Action states:

As to claim 7, the reference of Luu teaches of managing configurability of the application on a per-user basis (custom installation of application software on a user workstation, col. 5, lines 25-35).

Claim 7 is also rejected over the cited references. Claim 7 states,

7. The method of 6 wherein the differences between user files and differences between user registries may be used to manage configurability of the application on a per-user basis.

Examiner cites Luu at col. 5, lines 25-35, which state:

A second file utilized in the present invention is the personality file. The personality file allows for custom installation of application software on a user workstation. For example, if the application is to be installed in a particular directory, it is specified through the personality file. A custom personality file resides on the user workstation. In operation, the installation program on the user workstation will search for a custom personality file. If no custom personality file is found, a default personality file will be utilized to perform the installation. The personality file format is described in greater detail in appendix B.

This passage makes no mention of using differences between user files and user registries to manage configurability on a per-user basis. Not only does the passage not mention the differences (from initial to modified states) between user registries and differences between user files, it does not mention using such differences to manage configurability of the application on a per-user basis. It is therefore respectfully believed Examiner has failed to make out a prima facie case of obviousness under 35 USC 103. Claim 7 is therefore believed distinguished from the cited references.

II. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 12-9-02

Respectfully submitted,



Patrick C. R. Holmes
Reg. No. 46,380
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants

REDACTED CLAIMS

1. (ONCE AMENDED) A method for identifying and storing changes to a data processing system within a distributed data processing system, the method comprising the computer-implemented steps of:

initializing the data processing system for a capture of an initial state of the data processing system;

modifying the data processing system;

capturing [the] a modified state of the data processing system; and

storing differences between the initial state and the modified state as a set of configuration parameters in a depository, [wherein the set of configuration parameters may be used to manage configurability of a data processing system within the distributed data processing system.] wherein the differences are separated into system-specific changes and user-specific changes; and

wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis.

14. (ONCE AMENDED) An apparatus for identifying and storing changes to a data processing system within a distributed data processing system, the apparatus comprising:

initializing means for initializing the data processing system for a capture of an initial state of the data processing system;

modifying means for modifying the data processing system;

capturing means for capturing [the] a modified state of the data processing system; and

storing means for storing differences between the initial state and the modified state as a set of configuration parameters in a depository, [wherein the set of configuration parameters may be used to manage configurability of a data processing system within the distributed data processing system.] wherein the differences are separated into system-specific changes and user-specific changes; and

wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis.

27. (ONCE AMENDED) A computer program product on a computer-readable medium for identifying and storing changes to a data processing system within a distributed data processing system, the computer program product comprising:

- first instructions for initializing the data processing system for a capture of an initial state of the data processing system;
- second instructions for modifying the data processing system;
- third instructions for capturing [the] a modified state of the data processing system; and
- fourth instructions for storing differences between the initial state and the modified state as a set of configuration parameters in a depository, [wherein the set of configuration parameters may be used to manage configurability of a data processing system within the distributed data processing system.] wherein the differences are separated into system-specific changes and user-specific changes; and
wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis.

New Claims:

28. (NEW) A method of managing a distributed data processing system, comprising the computer implemented steps of:

- creating a package file which contains information indicating changes in a computer system from an initial state to a modified state;
- creating a user profile associated with a user;
- storing the package and user profile on a first computer of the distributed data processing system such that the package and user profile can be sent to individual computers of the distributed data processing system;
- upon startup of a second computer of the distributed data processing system, allowing access to the package and user profile by the second computer of the distributed data processing system.

29. (NEW) A method of managing a distributed data processing system, comprising the computer implemented steps of:

capturing changes in a computer system between an initial state and a modified state;

separating the changes into user-specific changes and system-specific changes;

wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis.